

# CASE 工具简介

CASE 是一种自动化或半自动化的方法，能够全面支持除系统调查外的每一个开发步骤。CASE 中集成了多种工具，这些工具既可以单独使用，也可以组合使用。随着软件工程的发展，CASE 的概念也由一种具体的工具发展成为开发信息系统的的方法学。

## CASE 工具分类

### 绘图工具

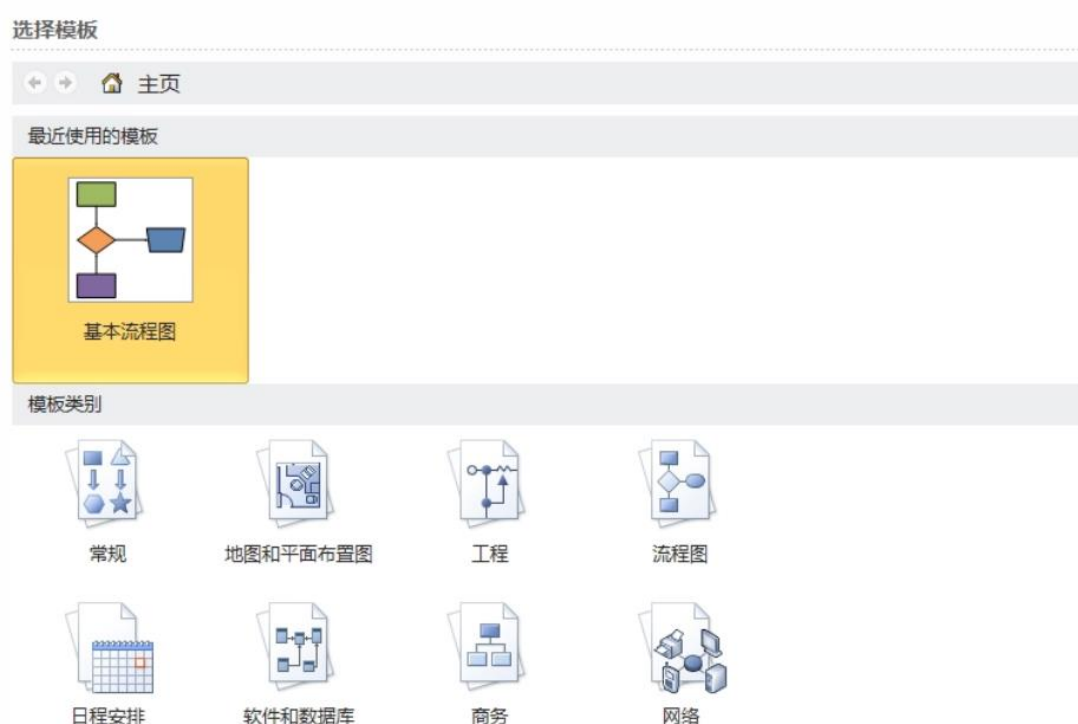
绘图工具用于以图形的形式表示系统组件、各种软件组件之间的数据和控制流以及系统结构。

### 流程建模工具

流程建模是创建软件过程模型的方法，被用来开发软件。流程建模工具帮助管理人员选择一个过程模型，或根据软件产品的要求修改它。

#### Microsoft Visio:

Visio 是一款由微软公司开发的专业流程图软件。它为用户提供了丰富的功能和工具，便于快速创建各种类型的流程图、图表、平面图和示意图。作为一款强大而直观的可视化工具，Visio 极大地简化了复杂流程的表达和可视化。



可以用于以下几个方面：

商务与项目管理：Visio 可用于绘制组织结构图、项目计划图、流程图等，帮助企业管理人员规划项目、优化流程，并进行效果展示和沟通。

网络拓扑与系统设计：Visio 提供了丰富的网络设备符号和模板，适用于绘制复杂的网络拓扑图和系统架构图，方便 IT 工程师进行网络规划和故障排查。

工业制图与流程优化：Visio 可用于绘制工艺流程图、流水线布局图等，帮助工厂和生产部门进行生产流程的可视化管理与优化，提高生产效率。

建筑与室内设计：Visio 拥有大量的平面图符号和 CAD 工具，可用于制作建筑布局图、家居设计图等，助力建筑师和设计师实现创意的可视化表达。


## 项目管理工具

项目管理工具被用于项目规划、成本和工作估算、项目调度和资源规划。管理人员必须严格遵守软件项目管理中提到的每个步骤的项目执行，该类工具有助于在

整个组织内实时存储和分享项目信息。

## Trac Project

Trac 是一个集成了 Wiki 和问题跟踪管理系统的应用平台，专为软件开发项目设计，是一个开源软件工具。



Integrated SCM & Project Management

搜索

登录 | 个人设置 | 帮助/指南 | 关于Trac

HomeTracTrac HacksGenshiBabelBitten

WIKI时间线路线图源码浏览查看任务单新建任务单搜索

最新变更修订日志版本库URL

缺省版本库

访问: 查看修订版: View diff against:

名称	大小	Rev	年龄	作者	最新变更
branches		17764	2个月	Jun Omae	1.6.1dev: use Subversion 1.14.3 for running tests on Windows [skip ci]
plugins		17767	3周	Dirk Stöcker	TracSpamFilter?: fix python 2/3 differences
sandbox		10651	13年	Eli Carter	sandbox: Ticket #3580 core change to make the set_owner permission ...
tags		17720	6个月	Ryan J Ollos	1.6: Release Trac 1.6 Refs #13333
trunk		17765	2个月	Jun Omae	1.7.1dev: merge [17764] from 1.6-stable [skip ci]

版本库索引

名称	大小	Rev	年龄	作者	最新变更
cboos.git		eb2df0ac	5年	Christian Boos	(#13146) add ticket comment buttons via insertNearReplyToComment ...
cboos.hg		8546:1951337739e2	11年	Christian Boos	TracGuide for 1.0.1: merge with -teo branch, for review.
jomae.git		dfa6c286	8个月	jun66j5	(#13456) use setuptools instead of distutils removed in Python 3.12
jomae.hg		12431:9077964163df	7年	jun66j5	(#11871) add unit tests for ...

主要可以用于几个方面：

项目文档管理：通过集成的 Wiki 系统，团队成员可以编写和共享项目相关的文档，如设计规范、用户手册等。

问题跟踪：Trac 提供了一个 Ticket 系统，用于记录、分类和跟踪软件开发过程中的问题、功能请求和其他任务。

版本控制集成：Trac 可以与版本控制系统（如 SVN）配合使用，实现代码管理和历史记录追踪。

时间轴追踪与产品路线图：帮助团队追踪项目进度，规划和可视化产品的发展方向。

代码浏览与搜索：支持原始码的浏览和搜索，方便开发人员查找和复用代码。

开发语言: Trac 是使用 Python 语言开发的, 因此在运行时需要 Python 环境的支持。

系统架构: 作为一个 Web 应用, Trac 旨在为软件开发团队提供一个集中的平台, 以支持项目管理和协作。它允许用户通过 Wiki 页面进行文档编写和共享, 同时提供问题跟踪系统来管理缺陷、功能请求和其他任务。

集成版本控制: Trac 可以与版本控制系统 (如 SVN) 配合使用, 提供文件管理、时间轴追踪、产品路线图、原始码浏览等功能。

安装要求: 在安装 Trac 时, 需要准备 Python 环境和相应的 Web 服务器 (如 Apache)。

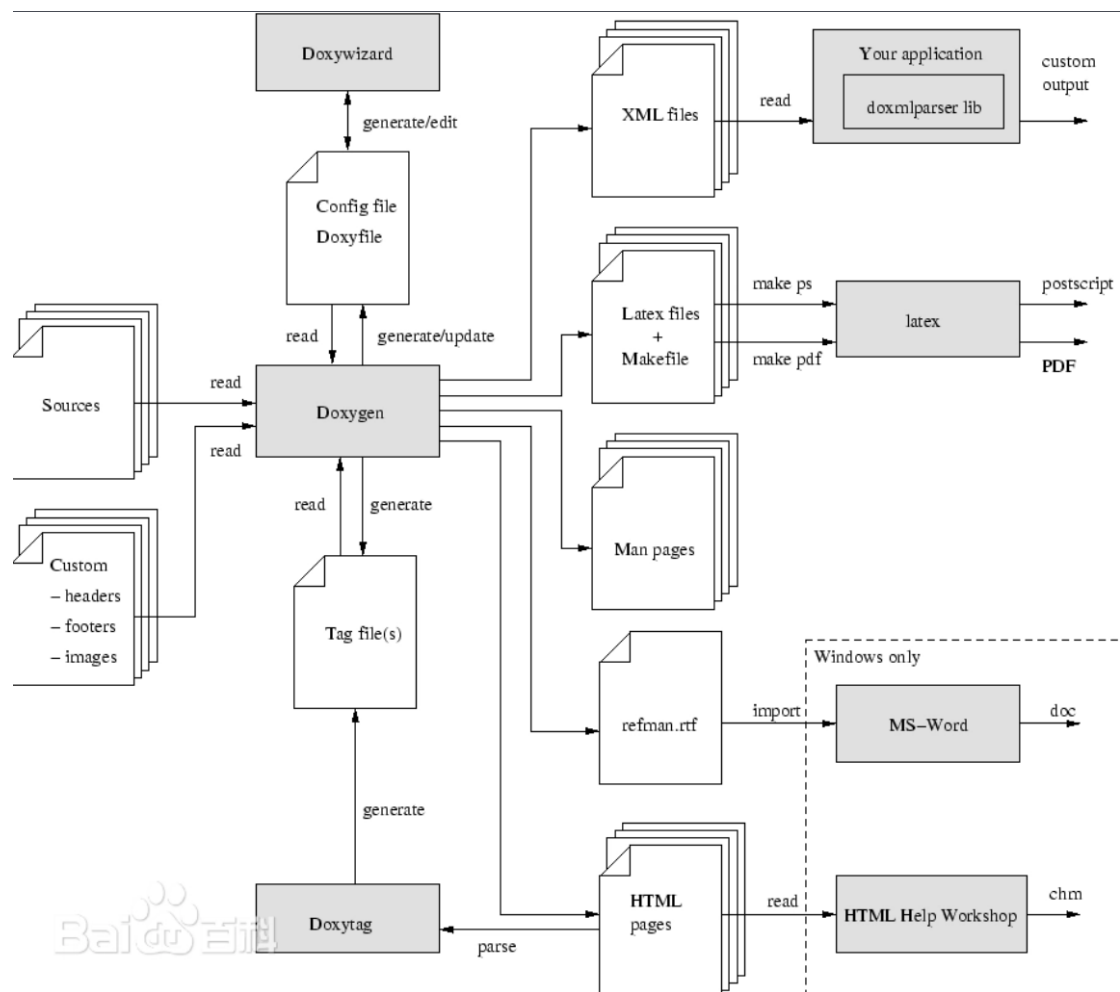
项目管理: Trac 适用于敏捷开发团队, 可以帮助团队成员更有效地协作和管理项目进度。

## 文档管理工具

文档管理工具为技术用户和终端用户生成文档。技术用户大多是开发团队的内部专业人员, 他们参考系统手册、参考手册、培训手册、安装手册等。终端用户的文件描述了系统的功能和如何操作, 如用户手册。

### Doxygen

Doxygen 是一种开源[跨平台](#)的, 以类似 [JavaDoc](#) 风格描述的[文档系统](#)。



Doxygen 是一个程序的文件产生工具，可将程序中的特定注释转换为说明文件。

对于未归档的[源文件](#)，也可以通过配置 Doxygen 来提取代码结构。

借助自动生成的包含依赖图（includedependency graphs）、继承图（inheritance diagram）以及[协作图](#)（collaborationdiagram）来可视化文档之间的关系。

Doxygen 生成的帮助文档的格式可以是 CHM、RTF、PostScript、PDF、HTML 和 [Unixman page](#) 等。

支持 C、C++、Java、[Objective-C](#) 和 [IDL](#) 语言，部分支持 [PHP](#)、C#。注释的语法与 [Qt-Doc](#)、KDoc 和 JavaDoc 兼容。

## 分析工具

分析工具有助于收集需求，自动检查任何不一致、不准确的图表、数据冗余或错误的遗漏。

## 设计工具

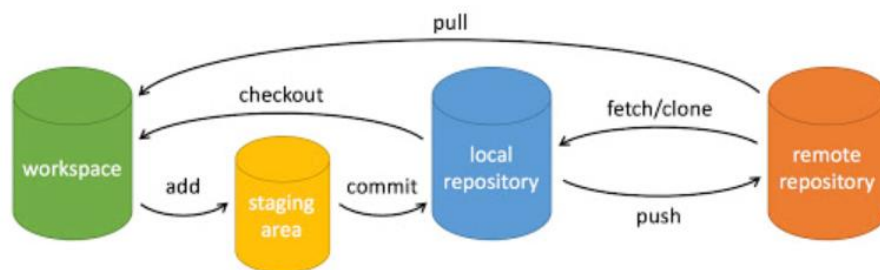
设计工具帮助软件设计者设计软件的块状结构，这些块状结构可以使用细化技术进一步分解成更小的模块。这些工具提供了每个模块的细节和模块间的相互联系。

## 配置管理工具

配置管理工具处理的是--版本和修订管理、基线配置管理、变更控制管理。

git:

Git 是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。



说明:

- workspace: 工作区
- staging area: 暂存区/缓存区
- local repository: 版本库或本地仓库
- remote repository: 远程仓库

关于 git 常用指令:

git init 初始化仓库

git add 添加文件到暂存区

git commit 提交暂存区到本地仓库

git push 上传远程代码并合并

git pull 下载远程代码并合并

git fetch/clone 从远程获取代码库

git branch 创建分支

git branch -d 删除分支

git merge 合并分支

git checkout 切换分支

git log 查看历史提交记录

## CVS:

如果说 Git 是一个分布式版本控制系统，那么 CVS 则是集中式的。这意味着 Git 允许开发者在本地进行版本控制操作，如提交、分支和合并等，然后再与远程仓库进行同步。相比之下，CVS 则依赖于中央服务器来管理版本控制，所有的版本历史都存储在服务器上，开发者需要从服务器获取和提交更改。

一个简单的 CVS 的工作流程：

关于 CVS 的安装与配置：

1.安装 CVS 服务器和客户端。

在 CVS 服务器上创建一个存储库（Repository），用于存储代码和文件。

2.检出代码：

使用 CVS 客户端从存储库中检出（Checkout）代码或文件到本地工作目录。

检出整个项目：cvs checkout dirname

检出特定文件：cvs checkout filename

3.修改代码：

在本地工作目录中对代码进行修改。

#### 4.更新代码：

在修改代码之前，通常建议先使用 CVS 客户端更新 (Update) 本地工作目录中的代码，以确保你正在修改的是最新的版本。

更新单个文件： `cvs update file_name`

更新整个目录： `cvs update` (不指定文件名时，将同步所有子目录下的文件)

#### 5.提交更改：

将修改后的代码通过 CVS 客户端提交 (Commit) 回存储库。

提交时，需要输入相关的日志信息，描述你所做的更改。

#### 6.解决冲突：

如果在提交过程中遇到版本冲突 (即多个开发者修改了同一部分代码)，需要使用 CVS 的冲突解决机制来合并代码，并重新提交。

#### 7.查看历史记录：

使用 CVS 客户端可以查看文件的历史记录 (Log)，了解文件的更改历史以及每个更改的详细信息。

#### 8.分支与合并：

对于大型项目或需要并行开发的场景，可能会使用到 CVS 的分支 (Branch) 和合并 (Merge) 功能。开发者可以在分支上进行特定的开发任务，完成后再将分支合并回主分支。

#### 其他操作

1.导入项目：使用 `cvs import` 命令将本地项目导入到 CVS 服务器。



2.导出项目：使用 `cvs checkout` 命令将项目从 CVS 服务器导出到本地工作目录。

需要注意的是，在使用 CVS 时，建议遵循“先同步，后修改”的原则，即在开始工作前，先更新本地工作目录中的文件到最新版本，再进行修改。此外，CVS 支持并发机制，允许多人同时修改同一文件，但需要注意合并冲突的问题。

## 编程工具

编程工具包括编程环境，如 IDE（集成开发环境）、内置模块库和仿真工具。这些工具为构建软件产品提供全面的帮助，并包括模拟和测试的功能。

### PyCharm

PyCharm 是由 JetBrains 公司推出的一款 Python 集成开发环境（IDE），为 Python 开发者提供了一套完整的工具，包括代码编辑器、调试器、测试工具、版本控制系统集成等。PyCharm 是一款功能强大、易于使用的 Python IDE，适合各种规模的 Python 项目开发和维护。

## 原型开发工具

原型开发工具可以创建独立于硬件的用户界面和设计。这些工具帮助我们在现有信息的基础上建立快速原型。此外，该类工具还提供软件原型的模拟。

### Serena prototype composer

Serena Prototype Composer 是一款面向非技术用户的软件策划、建模和原型设计工具。它允许用户可视化地定义软件需求，包括商业过程、活动、用户界面以及数据需求。它可以用于数据需求分析：支持对数据需求的分析和定义，这对于构建软件的数据层至关重要。

还可以用于以下几个方面：

可视化需求定义：它允许非技术用户通过可视化的方式定义软件需求，这包括商业过程、活动、用户界面和数据需求。这种可视化方法使得商业分析师等专业人士能够更容易地理解和表达软件的需求。

集成环境：提供了一个完整的集成环境，用户可以在其中进行设计和建模。这个环境旨在简化软件开发前的商业应用软件合理化过程。

模型驱动：支持由现有资源驱动的模式设计，这意味着用户可以基于现有的 web 应用软件等资源来创建模型。此外，它还支持将模型发布为可运行的原型和微软 Word 规格说明文档，这样利益相关者可以更直观地看到最终产品的外观和功能。

项目管理能力：除了界面原型设计，Serena Prototype Composer 还具备项目管理的功能，包括需求管理和数据管理。这使得它不仅可以用于设计阶段，还可以在整个项目管理过程中发挥作用。

国际化工具：虽然软件本身是英文版本，但它可以帮助国内项目经理提升产品设计和团队协调能力，尤其是在需要与国际团队合作的情况下。

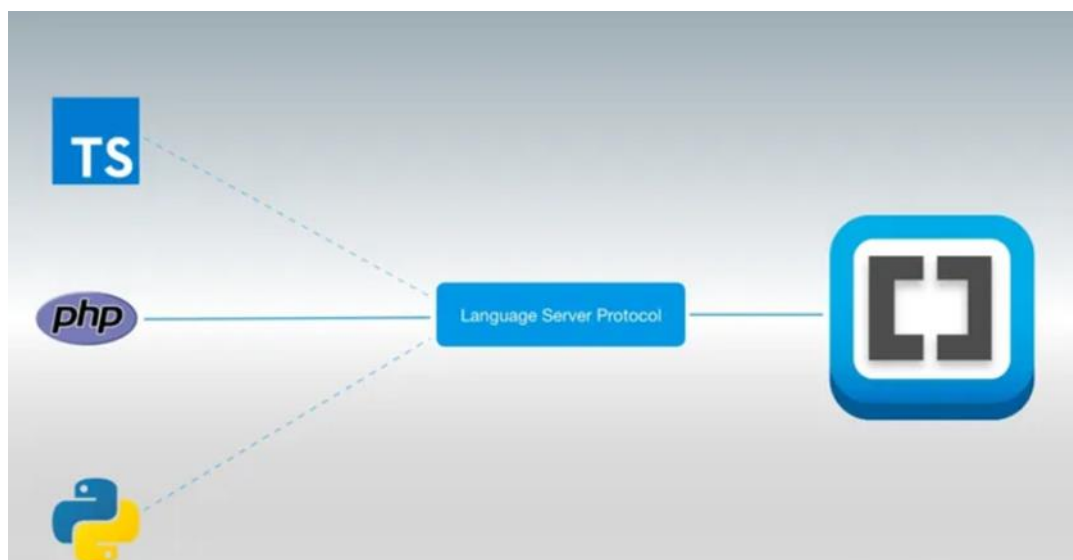
补充现有工具：对于已经熟悉并使用其他原型工具如 Axure 的用户来说，Serena Prototype Composer 可以作为一个补充工具，它的商业气息和 IT 集成气氛更加深厚，有助于更好地结合需求和设计。

## 网页开发工具

网页开发工具协助设计网页的所有相关元素，如表格、文本、脚本、图形等。网页开发工具还提供正在开发的内容和完成后的外观的实时预览。

Brackets:

Brackets是由 Adobe Systems 开发并在 GitHub 上发布的开源前端开发工具。这个工具可以免费使用，并完美兼容 Windows，Mac 以及 Linux 操作系统。Brackets 支持 HTML, CSS, JavaScript 三种通用语言，而且除了这三种语言之外，它还提供了 30 多种语言版本，几乎可以支持所有常见的语言。Brackets 内置的即时预览（Live Preview）功能，可以在编写网页时配合浏览器进行即时的更新，修改网页程式码后会同步显示于浏览器中，对于开发网页来说非常方便。此外，通过安装在 GitHub 上的扩展功能，用户还可以将其自定义为更顺手的前端工具。Brackets 的核心目标是减少在开发过程中那些效率低下的重复性工作，例如浏览器刷新，修改元素的样式，搜索功能等等。它简约、优雅、快捷，没有很多视图或者面板，也没太多花哨的功能，专门针对 WEB 前端开发设计。



## Adobe Edge Inspect

Adobe Edge Inspect 是一款专为移动开发者设计的工具，前身是 Adobe Shadow。它主要用于帮助设计师和开发者同时多个移动设备上预览应用设计，

从而发现并解决跨平台问题。通过 Adobe Edge Inspect，用户可以查看和调试手机设备中的文件，支持查看和修改 HTML，改变目标 CSS 以及进行 js 调试，同时支持 http 和 https 模式的认证。

## 质量保证工具

软件组织的质量保证是对开发软件产品的工程过程和方法进行监控，以确保质量符合组织的标准。质量保证工具包括配置工具以及软件测试工具。

### JMeter:

JMeter 是 Apache 组织开发的开源项目，是一个纯 Java 桌面应用，主要用于压力测试和性能测量。设计之初，它主要用于 Web 应用测试，但后来扩展到其它测试领域。有以下特性：

测试范围广泛：能够对 HTTP 和 FTP 服务器进行压力和性能测试，也可以对任何数据库进行同样的测试（通过 JDBC）。

可移植性与纯 Java 特性：JMeter 具有完全的可移植性，并且是 100% 的纯 Java 应用。

多线程与并发取样：它支持完全多线程框架，允许多个线程并发取样，并通过单独的线程组对不同的功能同时取样。

GUI 设计：其精心的 GUI 设计允许快速操作和更精确的计时。

结果分析与回放：JMeter 支持缓存和离线分析/回放测试结果。

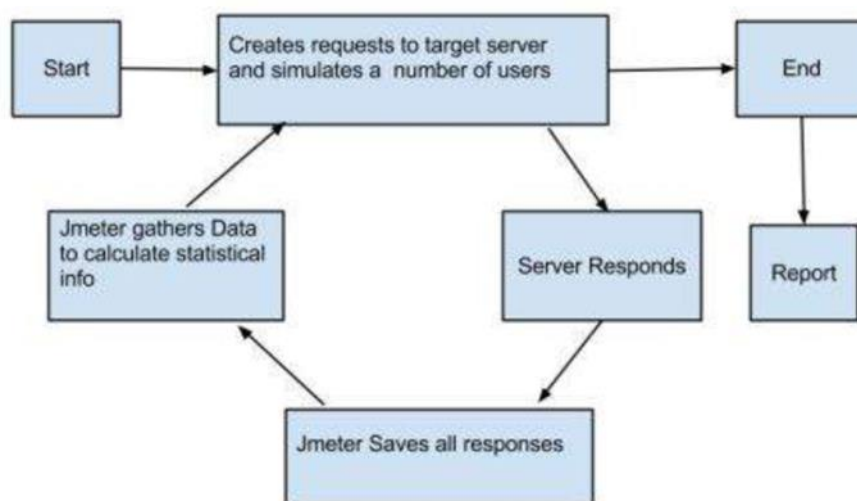
高可扩展性：JMeter 具有高度的可扩展性，链接的取样器允许无限制的测试能力。它还提供了各种负载统计表和可连接的计时器，数据分析和可视化插件提供了很好的可扩展性以及个性化。

功能/回归测试：JMeter 还能够对应用程序做功能/回归测试，通过创建带有断言

的脚本来验证程序返回了期望的结果。

HTTP 代理与请求记录：在设计阶段，JMeter 能够充当 HTTPPROXY（代理）来记录 IE/NETSCAPE 的 HTTP 请求，也可以记录 apache 等 WebServer 的 log 文件来重现 HTTP 流量。

下图描绘了 JMeter 的工作原理：



JMeter 由多个组件构成，如负载发生器、用户运行器、监听器、逻辑控制器、断言、配置元件、前置处理器和后置处理器、定时器和 HTTP 默认请求值等，每个组件都有其特定的功能和用途。这些组件使得 JMeter 在进行压力测试和性能测量时，能够模拟复杂的请求序列，收集和分析测试结果，从而帮助用户更好地理解 and 优化其应用的性能。

官方文档参考：[https://iowiki.com/jmeter/jmeter\\_quick\\_guide.html](https://iowiki.com/jmeter/jmeter_quick_guide.html)

## 维护工具

软件维护包括在软件产品交付后对其进行修改。自动记录和错误报告技术、自动错误单生成和根本原因分析是少数 CASE 工具，它们有助于软件组织在 SDLC 的维护阶段。例如, Bugzilla for defect tracking, HP Quality Center.